
Headers Migration Information

This document contains information for each header defined in XSH, Issue 5. Each section identifies changes made to the interface in Issue 5 (if any), complete with examples where appropriate. Only changes that might affect an application programmer are identified.

<aio.h>

Asynchronous input and output. (**REALTIME**)

```
#include <aio.h>
```

Issue 5: This is a new header included for alignment with the POSIX Realtime Extension.

The **<aio.h>** header defines the **aioctx** structure.

This header also includes the following constants:

```
AIO_CANCELED
AIO_NOTCANCELED
AIO_ALLDONE
LIO_WAIT
LIO_NOWAIT
LIO_READ
LIO_WRITE
LIO_NOP
```

This header includes declarations for the following functions: *aio_cancel()*, *aio_error()*, *aio_fsync()*, *aio_read()*, *aio_return()*, *aio_suspend()*, *aio_write()*, and *lio_listio()*.

<assert.h>

Verify program assertion.

```
#include <assert.h>
```

Issue 5: No functional changes in this issue.

<pio.h>

Cpio archive values.

```
#include <pio.h>
```

Issue 5: No functional changes in this issue.

<ctype.h>

Character types.

```
#include <ctype.h>
```

Issue 5: No functional changes in this issue.

<dirent.h>

Format of directory entries.

```
#include <dirent.h>
```

Issue 5: The Description is updated for alignment with the POSIX Threads Extension. Specifically, the thread-safe function *readdir_r()* is added.

<dlfcn.h>

Dynamic linking.

```
#include <dlfcn.h>
```

Issue 5: The **<dlfcn.h>** header defines the following macros for use in the *dlopen()* mode argument:

RTLD_LAZY	Relocations are performed at an implementation-dependent time.
RTLD_NOW	Relocations are performed when the object is loaded.
RTLD_GLOBAL	All symbols are available for relocation processing of other modules.
RTLD_LOCAL	All symbols are not made available for relocation processing by other modules.

This header includes declarations for the following functions: *dlopen()*, *dlsym()*, *dlclose()*, and *dlerror()*.

<errno.h>

System error numbers.

```
#include <errno.h>
```

Issue 5: Updated for alignment with the POSIX Realtime Extension. Specifically, the following error number is added:

[ECANCELED] Operation canceled.

<fcntl.h>

File control options.

```
#include <fcntl.h>
```

Issue 5: The Description is updated for alignment with the POSIX Realtime Extension. Specifically, the following changes are made:

File status flags used for *open()* and *fcntl()* are added:

O_DSYNC	Write according to synchronized I/O data integrity completion.
O_RSYNC	Synchronized read I/O operations.

<float.h>

Floating types.

```
#include <float.h>
```

Issue 5: No functional changes in this issue.

<fmtmsg.h>

Message display structures.

```
#include <fmtmsg.h>
```

Issue 5: No functional changes in this issue.

<fnmatch.h>

Filename-matching types.

```
#include <fnmatch.h>
```

Issue 5: No functional changes in this issue.

<ftw.h>

File tree traversal.

```
#include <ftw.h>
```

Issue 5: A description of FTW_DP is added:

FTW_DP	Directory with subdirectories visited.
--------	--

<glob.h>

Pathname pattern-matching types.

```
#include <glob.h>
```

Issue 5: No functional changes in this issue.

<grp.h>

Group structure.

```
#include <grp.h>
```

Issue 5: The Description is updated for alignment with the POSIX Threads Extension. Specifically, the thread-safe functions *getgrid_r()* and *getgrnam_r()* are added.

<iconv.h>

Codeset conversion facility.

```
#include <iconv.h>
```

Issue 5: No functional changes in this issue.

<inttypes.h>

Fixed size integral types.

```
#include <inttypes.h>
```

Issue 5: The **<inttypes.h>** header is a new header in the Single UNIX Specification, Version 2, included for data size neutrality.

This header includes definitions of at least the following types:

int16_t	16-bit signed integral type.
int32_t	32-bit signed integral type.
int64_t	64-bit signed integral type.
uint16_t	16-bit unsigned integral type.
uint32_t	32-bit unsigned integral type.
uint64_t	64-bit unsigned integral type.
intptr_t	Signed integral type large enough to hold any pointer.
uintptr_t	Unsigned integral type large enough to hold any pointer.

<iso646.h>

Alternative spellings.

```
#include <iso646.h>
```

Issue 5: This is a new header derived from the ISO/IEC 9899:1990/Amendment 1:1995 (E).

The **<iso646.h>** header defines the following eleven macros (on the left) that expand to the corresponding tokens (on the right):

and	&&
and_eq	&=
bitand	&
bitor	
compl	~
not	!
not_eq	!=
or	
or_eq	 =
xor	^
xor_eq	^=

<langinfo.h>

Language information constants.

```
#include <langinfo.h>
```

Issue 5: The constants YESSTR and NOSTR are marked LEGACY.

<libgen.h>

Definitions for pattern matching functions.

```
#include <libgen.h>
```

Issue 5: The function prototypes for *basename()* and *dirname()* are changed to indicate that the first argument is of type **char*** rather than **const char***.

<limits.h>

Implementation-dependent constants.

```
#include <limits.h>
```

Issue 5: The Description is updated for alignment with the POSIX Realtime Extension and the POSIX Threads Extension. Specifically, the following changes are made:

New **Runtime Invariant Values** are added.

AIO_LISTIO_MAX
AIO_MAX
AIO_PRIO_DELTA_MAX
DELAYTIMER_MAX
LOGIN_NAME_MAX
MQ_OPEN_MAX
MQ_PRIO_MAX
PTHREAD_DESTRUCTOR_ITERATIONS
PTHREAD_KEYS_MAX
PTHREAD_STACK_MIN
PTHREAD_THREADS_MAX
RTSIG_MAX
SEM_NSEMS_MAX
SEM_VALUE_MAX
SIGQUEUE_MAX
TIMER_MAX
TTY_NAME_MAX

New **Pathname Variable Values** are added:

FILESIZEBITS

FILESIZEBITS is a Large File Support extension.

A new **Maximum Value** is added:

_POSIX_CLOCKRES_MIN

New **Minimum Values** are added:

```

_POSIX_AIO_LISTIO_MAX
_POSIX_AIO_MAX
_POSIX_DELAYTIMER_MAX
_POSIX_MQ_OPEN_MAX
_POSIX_MQ_PRIO_MAX
_POSIX_RTSIG_MAX
_POSIX_SEM_NSEMS_MAX
_POSIX_SEM_VALUE_MAX
_POSIX_SIGQUEUE_MAX
_POSIX_THREAD_DESTRUCTOR_ITERATIONS
_POSIX_THREAD_KEYS_MAX
_POSIX_THREAD_THREADS_MAX
_POSIX_TIMER_MAX
_POSIX_TTY_NAME_MAX

```

The minimum acceptable values for `INT_MAX`, `INT_MIN`, and `UINT_MAX` are changed to make 32-bit values the minimum requirement.

<locale.h>

Category macros.

```
#include <locale.h>
```

Issue 5: No functional changes in this issue.

<math.h>

Mathematical declarations.

```
#include <math.h>
```

Issue 5: No functional changes in this issue.

<monetary.h>

Monetary types.

```
#include <monetary.h>
```

Issue 5: No functional changes in this issue.

<mqueue.h>

Message queues. (**REALTIME**)

```
#include <mqueue.h>
```

Issue 5: This is a new header included for alignment with the POSIX Realtime Extension.

The **<mqueue.h>** header defines the **mqd_t** type.

The **<mqueue.h>** header defines the **sigevent** structure (as described in **<signal.h>**) and the **mq_attr** structure.

This header includes declarations for the following functions: `mq_close()`, `mq_getattr()`, `mq_notify()`, `mq_open()`, `mq_receive()`, `mq_send()`, `mq_setattr()`, and `mq_unlink()`.

<ndbm.h>

Definitions for ndbm database operations.

```
#include <ndbm.h>
```

Issue 5: No functional changes are made, although explicit references to the definitions of **size_t** and **mode_t** are added to the Description.

<nl_types.h>

Data types.

```
#include <nl_types.h>
```

Issue 5: No functional changes in this issue.

<poll.h>

Definitions for the *poll()* function.

```
#include <poll.h>
```

Issue 5: No functional changes are made, although the definition of POLLIN is reworded:
POLLIN Same effect as POLLRDNORM | POLLRDBAND.

<pthread.h>

Threads.

```
#include <pthread.h>
```

Issue 5: This is a new header included for alignment with the POSIX Threads Extension.

The **<pthread.h>** header defines the following symbols:

```
PTHREAD_CANCEL_ASYNCHRONOUS
PTHREAD_CANCEL_ENABLE
PTHREAD_CANCEL_DEFERRED
PTHREAD_CANCEL_DISABLE
PTHREAD_CANCELED
PTHREAD_COND_INITIALIZER
PTHREAD_CREATE_DETACHED
PTHREAD_CREATE_JOINABLE
PTHREAD_EXPLICIT_SCHED
PTHREAD_INHERIT_SCHED
PTHREAD_MUTEX_DEFAULT
PTHREAD_MUTEX_ERRORCHECK
PTHREAD_MUTEX_NORMAL
PTHREAD_MUTEX_INITIALIZER
PTHREAD_MUTEX_RECURSIVE
PTHREAD_ONCE_INIT
PTHREAD_PRIO_INHERIT
PTHREAD_PRIO_NONE
PTHREAD_PRIO_PROTECT
PTHREAD_PROCESS_SHARED
PTHREAD_PROCESS_PRIVATE
PTHREAD_RWLOCK_INITIALIZER
PTHREAD_SCOPE_PROCESS
```

PTHREAD_SCOPE_SYSTEM

The **pthread_attr_t**, **pthread_cond_t**, **pthread_condattr_t**, **pthread_key_t**, **pthread_mutex_t**, **pthread_mutexattr_t**, **pthread_once_t**, **pthread_rwlock_t**, **pthread_rwlockattr_t**, and **pthread_t** types are defined as described in **<sys/types.h>**.

This header includes declarations for the following functions:

```
pthread_attr_destroy()
pthread_attr_getdetachstate()
pthread_attr_getguardsize()
pthread_attr_getinheritsched()
pthread_attr_getschedparam()
pthread_attr_getschedpolicy()
pthread_attr_getscope()
pthread_attr_getstackaddr()
pthread_attr_getstacksize()
pthread_attr_init()
pthread_attr_setdetachstate()
pthread_attr_setguardsize()
pthread_attr_setinheritsched()
pthread_attr_setschedparam()
pthread_attr_setschedpolicy()
pthread_attr_setscope()
pthread_attr_setstackaddr()
pthread_attr_setstacksize()
pthread_cancel()
pthread_cleanup_push()
pthread_cleanup_pop()
pthread_cond_broadcast()
pthread_cond_destroy()
pthread_cond_init()
pthread_cond_signal()
pthread_cond_timedwait()
pthread_cond_wait()
pthread_condattr_destroy()
pthread_condattr_getpshared()
pthread_condattr_init()
pthread_condattr_setpshared()
pthread_create()
pthread_detach()
pthread_equal()
pthread_exit()
pthread_getconcurrency()
pthread_getschedparam()
pthread_getspecific()
pthread_join()
pthread_key_create()
pthread_key_delete()
pthread_mutex_destroy()
pthread_mutex_getprioceiling()
pthread_mutex_init()
pthread_mutex_lock()
```



```
pthread_mutex_setprioceiling()  
pthread_mutex_trylock()  
pthread_mutex_unlock()  
pthread_mutexattr_destroy()  
pthread_mutexattr_getprioceiling()  
pthread_mutexattr_getprotocol()  
pthread_mutexattr_getpshared()  
pthread_mutexattr_gettype()  
pthread_mutexattr_init()  
pthread_mutexattr_setprioceiling()  
pthread_mutexattr_setprotocol()  
pthread_mutexattr_setpshared()  
pthread_mutexattr_settype()  
pthread_once()  
pthread_rwlock_destroy()  
pthread_rwlock_init()  
pthread_rwlock_rdlock()  
pthread_rwlock_tryrdlock()  
pthread_rwlock_trywrlock()  
pthread_rwlock_unlock()  
pthread_rwlock_wrlock()  
pthread_rwlockattr_destroy()  
pthread_rwlockattr_getpshared()  
pthread_rwlockattr_init()  
pthread_rwlockattr_setpshared()  
pthread_self()  
pthread_setcancelstate()  
pthread_setcanceltype()  
pthread_setconcurrency()  
pthread_setschedparam()  
pthread_setspecific()  
pthread_testcancel()
```

<pwd.h>

Password structure.

```
#include <pwd.h>
```

Issue 5: The Description is updated for alignment with the POSIX Threads Extension. Specifically, the thread-safe functions *getpwnam_r()* and *getpwuid_r()* are added.

<regex.h>

Regular-expression-matching types.

```
#include <regex.h>
```

Issue 5: No functional changes in this issue.

<re_comp.h>

Regular-expression-matching functions for *re_comp()*. (**LEGACY**)

```
#include <re_comp.h>
```

Issue 5: This header is marked LEGACY. New applications should use the *regcomp()*, *regexec()*, *regerror()*, and *regfree()* functions, and the **<regex.h>** header, which provide full internationalized regular expression functionality compatible with the POSIX.2 and XBD, Chapter 7, *Regular Expressions*.

<regex.h>

Regular-expression declarations. (**LEGACY**)

```
#include <regex.h>
```

Issue 5: This header is marked LEGACY. New applications should use the *regcomp()*, *regexec()*, *regerror()*, and *regfree()* functions, and the **<regex.h>** header, which provide full internationalized regular expression functionality compatible with the POSIX.2 and XBD, Chapter 7, *Regular Expressions*.

<sched.h>

Execution scheduling. (**REALTIME**)

```
#include <sched.h>
```

Issue 5: This is a new header included for alignment with the POSIX Realtime Extension.

The **<sched.h>** header defines the **sched_param** structure.

This header defines three standard scheduling policies indicated by the values of the following symbolic constants:

SCHED_FIFO	First in-first out (FIFO) scheduling policy.
SCHED_RR	Round robin scheduling policy.
SCHED_OTHER	Another scheduling policy.

This header includes declarations for the following functions:
sched_get_priority_max(), *sched_get_priority_min()*, *sched_getparam()*,
sched_getscheduler(), *sched_rr_get_interval()*, *sched_setparam()*,
sched_setscheduler(), and *sched_yield()*

<search.h>

Search tables.

```
#include <search.h>
```

Issue 5: No functional changes in this issue.

<semaphore.h>

Semaphores. (**REALTIME**)

```
#include <semaphore.h>
```

Issue 5: This is a new header included for alignment with the POSIX Realtime Extension.

The **<semaphore.h>** header defines the **sem_t** type.

This header includes declarations for the following functions: *sem_close()*, *sem_destroy()*, *sem_getvalue()*, *sem_init()*, *sem_open()*, *sem_post()*, *sem_trywait()*, *sem_unlink()*, and *sem_wait()*.

<setjmp.h>

Stack environment declarations.

```
#include <setjmp.h>
```

Issue 5: No functional changes in this issue.

<signal.h>

Signals.

```
#include <signal.h>
```

Issue 5: The Description is updated for alignment with the POSIX Realtime Extension and the POSIX Threads Extension. Specifically:

The **<signal.h>** header defines the **sigevent** structure.

The **sigval** union is defined.

This header also declares the macros SIGRTMIN and SIGRTMAX.

This header includes the following new function declarations: *pthread_kill()*, *pthread_sigmask()*, *sigqueue()*, *sigtimedwait()*, *sigwaitinfo()*, and *sigwait()*.

The default action for SIGURG is changed from i to iii. The function prototype for *sigmask()* is removed.

<stdarg.h>

Handle variable argument list.

```
#include <stdarg.h>
```

```
void va_start(va_list ap, argN);  
type va_arg(va_list ap, type);  
void va_end(va_list ap);
```

Issue 5: No functional changes in this issue.

<stddef.h>

Standard type definitions.

```
#include <stddef.h>
```

Issue 5: No functional changes in this issue.

<stdio.h>

Standard buffered input/output.

```
#include <stdio.h>
```

Issue 5: The Description is updated for alignment with the POSIX Threads Extension. Specifically, the *flockfile()*, *ftrylockfile()*, *funlockfile()*, *getc_unlocked()*, *getchar_unlocked()*, *putc_unlocked()*, and *putchar_unlocked()* functions are added.

Large File Support extensions are added; specifically, the declarations of the *fseeko()* and *ftello()* functions.

The constant *L_cuserid* and the external variables *optarg*, *opterr*, *optind*, and *optopt* are marked as extensions and LEGACY.

The *cuserid()* and *getopt()* functions are marked LEGACY.

<stdlib.h>

Standard library definitions.

```
#include <stdlib.h>
```

Issue 5: The Description is updated for alignment with the POSIX Threads Extension. Specifically, the *rand_r()* function is added.

The *ttyslot()* and *valloc()* functions are marked LEGACY.

The type of the third argument to *initstate()* is changed from **int** to **size_t**. The type of the return value from *setstate()* is changed from **char** to **char***, and the type of the first argument is changed from **char*** to **const char***.

<string.h>

String operations.

```
#include <string.h>
```

Issue 5: The Description is updated for alignment with the POSIX Threads Extension. Specifically, the *strtok_r()* function is added.

<strings.h>

String operations.

```
#include <strings.h>
```

Issue 5: No functional changes in this issue.

<stropts.h>

STREAMS interface.

```
#include <stropts.h>
```

Issue 5: The *flags* member of the **strpeek** and **strfdinsert** structures is changed from type **long** to type **t_uscalar_t**.

<syslog.h>

Definitions for system error logging.

```
#include <syslog.h>
```

Issue 5: No functional changes in this issue.

<sys/ipc.h>

Interprocess communication access structure.

```
#include <sys/ipc.h>
```

Issue 5: No functional changes in this issue.

<sys/mman.h>

Memory management declarations.

```
#include <sys/mman.h>
```

Issue 5: Updated for alignment with the POSIX Realtime Extension; specifically:

The following symbolic constants are defined for the *mlockall()* function:

MCL_CURRENT	Lock currently mapped pages.
MCL_FUTURE	Lock pages that become mapped.

The symbolic constant MAP_FAILED is defined to indicate a failure from the *mmap()* function.

This header includes new declarations for the following functions: *mlock()*, *mlockall()*, *munlock()*, *munlockall()*, *shm_open()*, and *shm_unlink()*.

<sys/msg.h>

Message queue structures.

```
#include <sys/msg.h>
```

Issue 5: No functional changes in this issue.

<sys/resource.h>

Definitions for XSI resource operations.

```
#include <sys/resource.h>
```

Issue 5: Large File Support extensions are added. Specifically, the following changes are made:

New symbolic constants are defined:

RLIM_SAVED_MAX	A value of type rlim_t indicating an unrepresentable saved
----------------	---

hard limit.
 RLIM_SAVED_CUR A value of type **rlim_t** indicating an unrepresentable saved soft limit.

On implementations where all resource limits are representable in an object of type **rlim_t**, RLIM_SAVED_MAX and RLIM_SAVED_CUR need not be distinct from RLIM_INFINITY.

<sys/sem.h>

Semaphore facility.

```
#include <sys/sem.h>
```

Issue 5: No functional changes in this issue.

<sys/shm.h>

Shared memory facility.

```
#include <sys/shm.h>
```

Issue 5: The element of *shm_segsz* specifies the size of a memory segment. This has been changed for data size neutrality. The type of *shm_segsz* is changed from **int** to **size_t**.

<sys/stat.h>

Data returned by the *stat()* function.

```
#include <sys/stat.h>
```

Issue 5: The Description is updated for alignment with the POSIX Realtime Extension. Specifically, the following macros are added:

S_TYPEISMQ(*buf*) Test for a message queue.
 S_TYPEISSEM(*buf*) Test for a semaphore.
 S_TYPEISSHM(*buf*) Test for a shared memory object.

The following Large File Support extension is made:

The type of *st_blksize* is changed from **long** to **blksize_t**; the type of *st_blocks* is changed from **long** to **blkcnt_t**.

<sys/statvfs.h>

VFS Filesystem information structure.

```
#include <sys/statvfs.h>
```

Issue 5: The following Large File Support extension is made:

The type of *f_blocks*, *f_bfree*, and *f_bavail* is changed from **unsigned long** to **fsblkcnt_t**; the type of *f_files*, *f_ffree*, and *f_favail* is changed from **unsigned long** to **fsfilcnt_t**.

<sys/time.h>

Time types.

```
#include <sys/time.h>
```

Issue 5: The type of *tv_usec* is changed from **long** to **suseconds_t**, which is a signed arithmetic type used for time in microseconds.

<sys/timeb.h>

Additional definitions for date and time.

```
#include <sys/timeb.h>
```

Issue 5: No functional changes in this issue.

<sys/times.h>

File access and modification times structure.

```
#include <sys/times.h>
```

Issue 5: No functional changes in this issue.

<sys/types.h>

Data types.

```
#include <sys/types.h>
```

Issue 5: The **clockid_t** and **timer_t** types are defined for alignment with the POSIX Realtime Extension.

Large File Support extensions are added; specifically, the types **blkcnt_t**, **blksize_t**, **fsblkcnt_t**, **fsfilcnt_t**, and **suseconds_t** are added.

Updated for alignment with the POSIX Threads Extension and the X/Open Threads Extension. Specifically, the **pthread_attr_t**, **pthread_cond_t**, **pthread_condattr_t**, **pthread_key_t**, **pthread_mutex_t**, **pthread_mutexattr_t**, **pthread_once_t**, **pthread_rwlock_t**, **pthread_rwlockattr_t**, and **pthread_t** types are added.

<sys/uio.h>

Definitions for vector I/O operations.

```
#include <sys/uio.h>
```

Issue 5: No functional changes in this issue.

<sys/utsname.h>

System name structure.

```
#include <sys/utsname.h>
```

Issue 5: No functional changes in this issue.

<sys/wait.h>

Declarations for waiting.

```
#include <sys/wait.h>
```

Issue 5: No functional changes in this issue.

<tar.h>

Extended *tar* definitions.

```
#include <tar.h>
```

Issue 5: No functional changes in this issue.

<termios.h>

Define values for *termios*.

```
#include <termios.h>
```

Issue 5: No functional changes in this issue.

<time.h>

Time types.

```
#include <time.h>
```

Issue 5: The Description is updated for alignment with the POSIX Realtime Extension and the POSIX Threads Extension. Specifically, the following changes are made:

The **<time.h>** header declares the structure **timespec**.

This header also declares the **itimerspec** structure.

The following constants are defined:

CLOCK_REALTIME The identifier of the system-wide realtime clock.

TIMER_ABSTIME Flag indicating time is absolute with respect to the clock associated with a timer.

The following new function declarations are added: *asctime_r()*, *clock_getres()*, *clock_gettime()*, *clock_settime()*, *ctime_r()*, *gmtime_r()*, *localtime_r()*, *nanosleep()*, *timer_create()*, *timer_delete()*, *timer_gettime()*, *timer_getoverrun()*, and *timer_settime()*.

<ucontext.h>

User context.

```
#include <ucontext.h>
```

Issue 5: No functional changes in this issue.

<ulimit.h>

Ulimit commands.

```
#include <ulimit.h>
```

Issue 5: No functional changes in this issue.

<unistd.h>

Standard symbolic constants and types.

```
#include <unistd.h>
```

Issue 5: The Description is updated for alignment with the POSIX Realtime Extension and the POSIX Threads Extension.

The following symbolic constants are always defined to unspecified values to indicate that this functionality from the POSIX Threads Extension is always present on XSI-conformant systems:

```
_POSIX_THREADS  
_POSIX_THREAD_ATTR_STACKADDR  
_POSIX_THREAD_ATTR_STACKSIZE  
_POSIX_THREAD_PROCESS_SHARED  
_POSIX_THREAD_SAFE_FUNCTIONS
```

The symbolic constants `_XOPEN_REALTIME` and `_XOPEN_REALTIME_THREADS` are added. `_POSIX2_C_BIND`, `_XOPEN_ENH_I18N`, and `_XOPEN_SHM` must now be set to a value other than `-1` by a conforming implementation.

If `_XOPEN_REALTIME` is defined to have a value other than `-1`, then the following symbolic constants will be defined to an unspecified value to indicate that the features are supported:

```
_POSIX_ASYNCHRONOUS_IO  
_POSIX_MEMLOCK  
_POSIX_MEMLOCK_RANGE  
_POSIX_MESSAGE_PASSING  
_POSIX_PRIORITY_SCHEDULING  
_POSIX_REALTIME_SIGNALS  
_POSIX_SEMAPHORES  
_POSIX_SHARED_MEMORY_OBJECTS  
_POSIX_SYNCHRONIZED_IO  
_POSIX_TIMERS
```

The following symbolic constants are always defined to unspecified values to indicate that the functionality is always present on XSI-conformant systems:

```
_POSIX_FSYNC  
_POSIX_MAPPED_FILES  
_POSIX_MEMORY_PROTECTION
```

The following symbolic constant will be defined if the option is supported; otherwise, it will be undefined:

```
_POSIX_PRIORITIZED_IO
```

If `_XOPEN_REALTIME_THREADS` is defined to have a value other than `-1`, then the following symbolic constants will be defined to an unspecified value to indicate that the features are supported:

```
_POSIX_THREAD_PRIORITY_SCHEDULING
_POSIX_THREAD_PRIO_INHERIT
_POSIX_THREAD_PRIO_PROTECT
```

Large File Support extensions added.

The type of the argument to `sbrk()` is changed from `int` to `intptr_t`.

`_XBS_` constants are added to the list of Constants for Options and Feature Groups, to the list of constants for the `confstr()` function, and to the list of constants to the `sysconf()` function. These are to support the different programming models:

```
_XBS5_ILP32_OFF32
_XBS5_ILP32_OFFBIG
_XBS5_LP64_OFF64
_XBS5_LPBIG_OFFBIG
```

The following new symbolic constants are defined for the `confstr()` function:

```
_CS_XBS5_ILP32_OFF32_CFLAGS
_CS_XBS5_ILP32_OFF32_LDFLAGS
_CS_XBS5_ILP32_OFF32_LIBS
_CS_XBS5_ILP32_OFF32_LINTFLAGS
_CS_XBS5_ILP32_OFFBIG_CFLAGS
_CS_XBS5_ILP32_OFFBIG_LDFLAGS
_CS_XBS5_ILP32_OFFBIG_LIBS
_CS_XBS5_ILP32_OFFBIG_LINTFLAGS
_CS_XBS5_LP64_OFF64_CFLAGS
_CS_XBS5_LP64_OFF64_LDFLAGS
_CS_XBS5_LP64_OFF64_LIBS
_CS_XBS5_LP64_OFF64_LINTFLAGS
_CS_XBS5_LPBIG_OFFBIG_CFLAGS
_CS_XBS5_LPBIG_OFFBIG_LDFLAGS
_CS_XBS5_LPBIG_OFFBIG_LIBS
_CS_XBS5_LPBIG_OFFBIG_LINTFLAGS
```

The following new symbolic constants are defined for the `sysconf()` function:

```
_SC_AIO_LISTIO_MAX
_SC_AIO_MAX
_SC_AIO_PRIO_DELTA_MAX
_SC_ASYNCHRONOUS_IO
_SC_DELAYTIMER_MAX
_SC_FSYNC
_SC_GETGR_R_SIZE_MAX
_SC_GETPW_R_SIZE_MAX
_SC_MAPPED_FILES
_SC_MEMLOCK
_SC_MEMLOCK_RANGE
_SC_MEMORY_PROTECTION
_SC_MESSAGE_PASSING
_SC_MQ_OPEN_MAX
```

```
_SC_MQ_PRIO_MAX
_SC_PRIORITIZED_IO
_SC_PRIORITY_SCHEDULING
_SC_REALTIME_SIGNALS
_SC_RTSIG_MAX
_SC_SEMAPHORES
_SC_SEM_NSEMS_MAX
_SC_SEM_VALUE_MAX
_SC_SHARED_MEMORY_OBJECTS
_SC_SIGQUEUE_MAX
_SC_SYNCHRONIZED_IO
_SC_THREADS
_SC_THREAD_ATTR_STACKADDR
_SC_THREAD_ATTR_STACKSIZE
_SC_THREAD_DESTRUCTOR_ITERATIONS
_SC_THREAD_KEYS_MAX
_SC_THREAD_PRIORITY_SCHEDULING
_SC_THREAD_PRIO_INHERIT
_SC_THREAD_PRIO_PROTECT
_SC_THREAD_PROCESS_SHARED
_SC_THREAD_SAFE_FUNCTIONS
_SC_THREAD_STACK_MIN
_SC_THREAD_THREADS_MAX
_SC_TIMERS
_SC_TIMER_MAX
_SC_XBS5_ILP32_OFF32
_SC_XBS5_ILP32_OFFBIG
_SC_XBS5_LP64_OFF64
_SC_XBS5_LPBIG_OFFBIG
```

The following new symbolic constants are defined for the *pathconf()* and *fpathconf()* functions:

```
_PC_ASYNC_IO
_PC_SYNC_IO
_PC_PRIO_IO
```

The following new function declarations are added to this header: *getlogin_r()*, *pread()*, *pthread_atfork()*, *pwrite()*, and *ttyname_r()*.

<utime.h>

Access and modification times structure.

```
#include <utime.h>
```

Issue 5: No functional changes in this issue.

<utmpx.h>

User accounting database definitions.

```
#include <utmpx.h>
```

Issue 5: No functional changes in this issue.

<varargs.h>

Handle variable argument list. (**LEGACY**)

```
#include <varargs.h>
```

```
va_alist
va_dcl
void va_start(pvar)
va_list pvar;
type va_arg(pvar, type)
va_list pvar;
void va_end(pvar)
va_list pvar;
```

Issue 5: This header is marked LEGACY. New applications should use **<stdarg.h>**.

<wchar.h>

Wide-character types.

```
#include <wchar.h>
```

Issue 5: Aligned with the ISO/IEC 9899:1990/Amendment 1:1995 (E). Specifically, the prototypes for *putwc()*, *wcsftime()*, and *wcstok()* have been changed in this issue.

The *wcswcs()* function has been marked as an Extension. New applications should use the *wcsstr()* function instead.

<wctype.h>

Wide-character classification and mapping utilities.

```
#include <wctype.h>
```

Issue 5: This header is a new header derived from the ISO/IEC 9899:1990/Amendment 1:1995 (E).

The **<wctype.h>** header defines the following data types through **typedef**:

wint_t	As described in <wchar.h> .
wctrans_t	A scalar type that can hold values which represent locale-specific character mappings.
wctype_t	As described in <wchar.h> .

This header includes declarations for the following functions: *iswalnum()*, *iswalpha()*, *iswcntrl()*, *iswdigit()*, *iswgraph()*, *iswlower()*, *iswprint()*, *iswpunct()*, *iswspace()*, *iswupper()*, *iswxdigit()*, *iswctype()*, *towctrans()*, *towlower()*, *towupper()*, *wctrans()*, and *wctype()*.

<wordexp.h>

Word-expansion types.

```
#include <wordexp.h>
```

Issue 5: No functional changes in this issue.